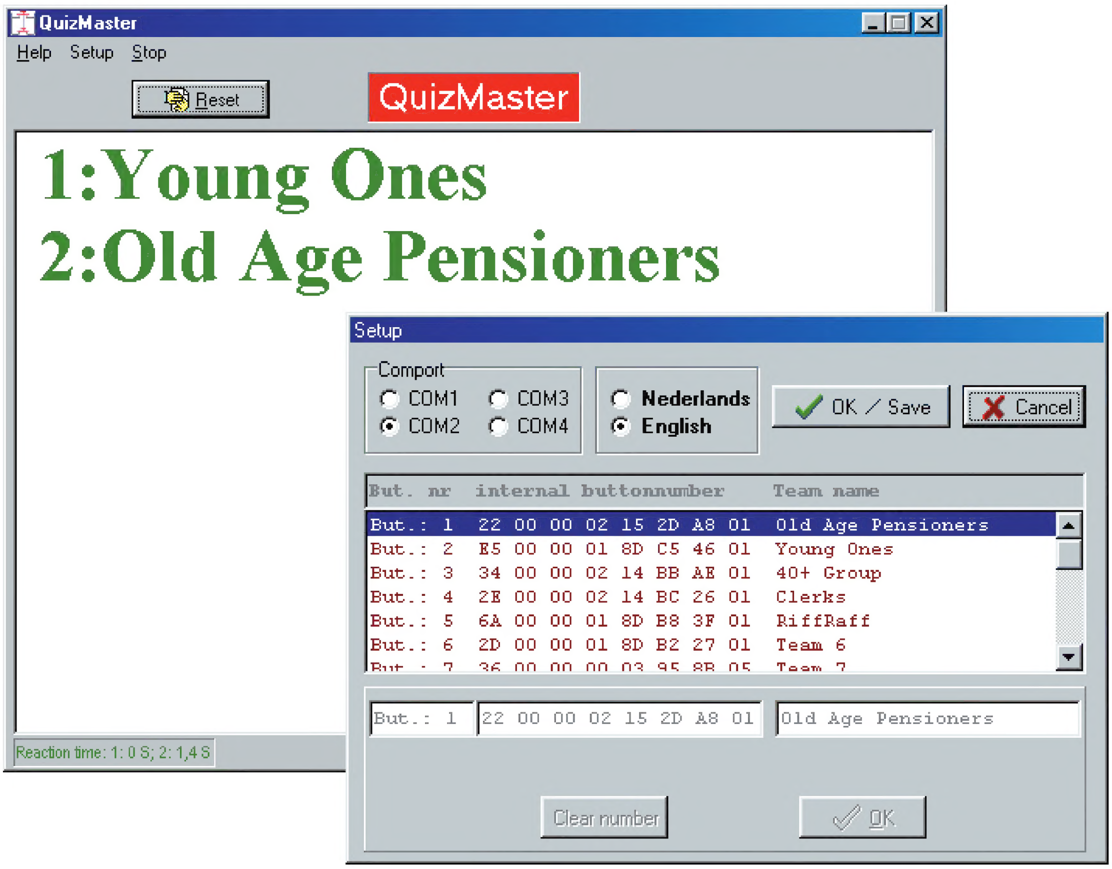
QuizMaster

the 1-wire model

Design by T. Rudolphi

With some simple hardware, ditto Windows program and a PC, it is possible to determine which of up to 20 switches is pressed first. Even the exact order can be displayed on a PC monitor. Ideal for quizzes and other games!



```
1:Young Ones
2:Old Age Pensioners

Reaction time: 1: 0 S; 2: 1,4 S
```

Comport: ○ COM1 ● COM2 ○ COM3 ○ COM4

○ Nederlands ● English

```
But. nr  internal buttonnumber     Team name
But.: 1  22 00 00 02 15 2D A8 01   Old Age Pensioners
But.: 2  E5 00 00 01 8D C5 46 01   Young Ones
But.: 3  34 00 00 02 14 BB AE 01   40+ Group
But.: 4  2E 00 00 02 14 BC 26 01   Clerks
But.: 5  6A 00 00 01 8D B8 3F 01   RiffRaff
But.: 6  2D 00 00 01 8D B2 27 01   Team 6
But.: 7  36 00 00 00 03 95 8B 05   Team 7

But.: 1  22 00 00 02 15 2D A8 01   Old Age Pensioners
```

The most remarkable feature of this QuizMaster is its simplicity. In its entirety, it consists of no more than a trivial interface, a number of pushbuttons and a Windows program. The interface comprises, all in all, seven components and is connected to the COM-port of a PC. The pushbuttons for the quiz contestants (a maximum of 20) are fitted with a '1-wire multidrop'-device and are all connected in parallel to the interface. The program, written in 'C++ Builder' from Borland, controls the entire system and displays on the PC screen the order in which the pushbuttons were pressed.

1-Wire chips

The essential part for the operation of the circuit described here, is the ‘1-wire-chip’, which is fitted into each pushbutton. This type of device has already featured in Elektor Electronics on earlier occasions, such as the articles ‘1-Wire Spy’, published in June 2000 and ‘e-Key’ in the November 2000 issue. However, for completeness’ sake, a brief description is included here.

For some time now, Dallas Semiconductor have been offering a range of components, which can be controlled, read and programmed by a master (a microcontroller for example) via a 1-wire bus. These ‘1-wire-devices’ are specifically intended for tasks such as temperature measurement, battery management and data storage. With this 1-wire protocol, an integrated multiplexer unravels the incoming signals and passes the correct ones on to the chip. Because the output of a 1-wire device is open-drain, multiple devices are connected to a single bus in Wired-And configuration. It is also possible to mix these devices with ‘normal’ hardware.

Although the capabilities of the available 1-wire devices far exceed the requirements of the present application, the relatively reasonably priced DS2401 is nevertheless very appropriate for our use. This device can best be described as an identifier IC that has no other special function but functions solely as an ‘electronic registration number’. The chip contains a 64-bit ‘lasered’ ROM containing a unique 48-bit serial number, an 8-bit CRC and an 8-bit family code. Reading the device allows immediate recognition and identification — and that is exactly what we need here.

Every 1-wire-chip possesses the identification feature and is, without exception, suitable for our application. For example, the DS1820 temperature sensor is readily available but is twice as expensive compared to the DS2401, and it is a pity of course, that its primary functionality would not be utilised.

Hardware

The schematic for the required hardware is shown in **Figure 1**. Shown on the left is the PC-interface and on the right a pushbutton with a 1-wire device, of which up to 20 may be connected in parallel on the 1-wire-bus (terminals DATA and GND). Whenever the pushbutton (S1) is pressed, the 1-wire-device is connected to the bus and can be recognised by the PC.

From the electronic perspective, simplicity is trumps. The interface consists of no more than a couple of (zener) diodes, two resistors and a capacitor while the control unit contains no components apart from the pushbutton and the IC.

The interface is powered from the PC via the RS232 connector. Resistor R2 and electrolytic capacitor C1 limit and filter the sup-

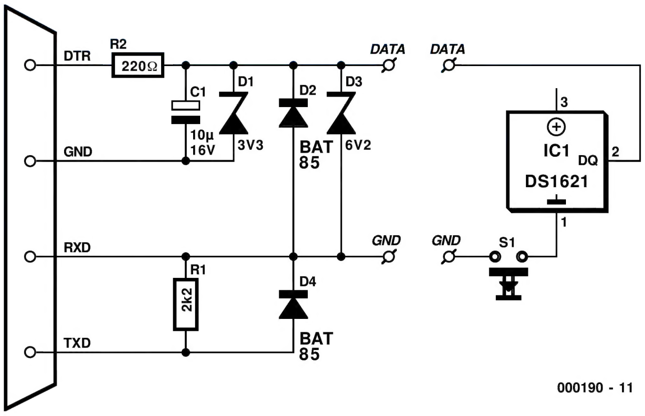


Figure 1. The hardware consists of an interface for the RS232-port and a pushbutton with a 1-wire-device.

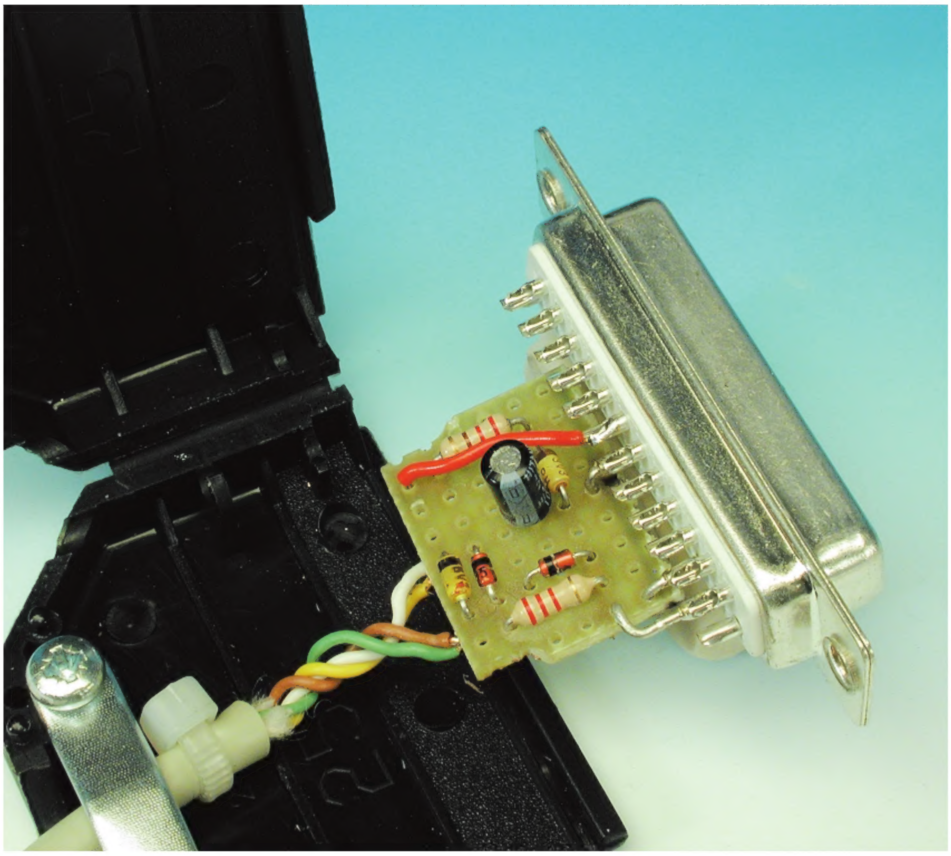


Figure 2. The entire interface can easily be built into an RS232 connector.

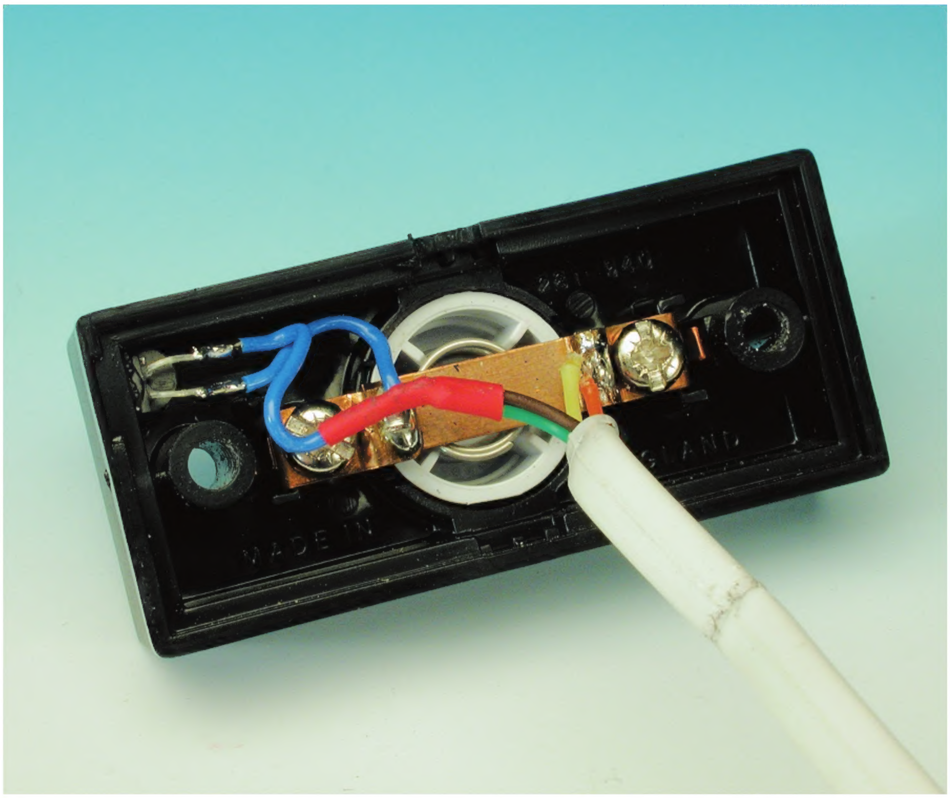


Figure 3. Inside this pushbutton is plenty of room for the DS2401.

ply voltage. Also, because the current requirement of the 1-wire devices doesn't amount to more than a few milliamps, pull-up resistor R1 limits the current to about 6 mA.

The length of cable between the interface and the pushbutton(s) should not exceed about 30 meters.

Simple Soldering Job

The interface is of such a simple design that it is not worth the effort to put it in a separate enclosure. It is much easier to connect the few components directly 'point-to-point' or on a small piece of prototyping board inside a sub-D connector. These connectors offer ample space as is illustrated by the photo of the prototype, shown in **Figure 2**. When a 25-way RS-232 connector is used, DTR, GND, RXD and TXD are pins 20, 7, 3 and 2 respectively. With a 9-way connector these are pins 4, 5, 2 and 3 respectively.

Because it is nice for the (nervous) contestant to have a pushbutton that is comfortable to the hand, we picked a generously sized doorbell button, which was secured to a small block of wood for the occasion. The DS2401 can be fastened on the inside with a drop of glue. Again, **Figure 3** shows how the prototype was constructed. The GND connection can be looped directly to one side of the pushbutton. The DS2401 is connected in series with the DATA line.

Thus prepared, all the candidates' pushbuttons have to be connected in parallel to the 1-wire-bus. The easiest way to do this is to use RJ-11 style telephone connectors (6-way with 4 contacts), which may be obtained from any DIY store. These connectors can be fitted to a flat telephone cable with the aid of a crimping tool. Using adapters with three female connectors it is possible to interconnect all the switches. The connections are as follows:

pin 1 = nc,
pin 2 = GND,
pin 3 = DATA,
pin 4 = DATA,
pin 5 = GND, and
pin 6 = nc.

This may appear a little more complicated than necessary, but the double connection prevents the crossover in the three-way adapter from throwing a spanner in the works.

Software

The software has been developed with 'C++ Builder' from Borland and runs under Windows 95/98. The program, including the source code, can be purchased through Elektor Electronics Readers Services (Order code **000190-11**) or may be downloaded for free from our web site.

A brief description of the program is as follows:

- One thread continually sends a 1-wire reset (a 480 µs low signal, followed by a 'listen-high period' of 480 µs).
- When a pushbutton is pressed, the built-in IC will respond to this signal.
- Then, the internal 64-bit identification number is retrieved, to see which switch was activated. An init file links this number with a contestant and this is what is displayed on the screen.

Setup

Before the QuizMaster can be put to use, a number of things need to be initialised. First, the RS232 (COM) port has to be selected. This can be done from the welcome screen.

If the interface is connected properly and the port is configured correctly, it is possible to select from the main screen the 'setup' menu item. A table then displays all 20 pushbuttons. For each of these pushbuttons it is necessary to know the internal 64-bit number and the name of the team (name of the contestant).

- To enter on the team name, double click the desired switch in the list. The underlying edit screen is activated and the name can then be entered in the last field.

- The 64-bit number has to be learned. When the desired quiz button is pressed, the corresponding number will appear in the appropriate field. After which you only need to click OK.

Repeat the above procedure for each of the switches. When everything is complete, click 'OK and Save'. All the information is now stored in a config.txt file (ASCII format), from which the program obtains the information when starting up. The QuizMaster is now ready for use!

(000190-1)